

Oracle Database Technology Night

～集え！オラクルの力(チカラ)～

DB 12cから実装された
マルチテナント・アーキテクチャで
DBがより使いやすくなる

日本オラクル株式会社
クラウド・テクノロジー事業統括
Database & Exadataプロダクトマネジメント本部

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Technical Discussion Night

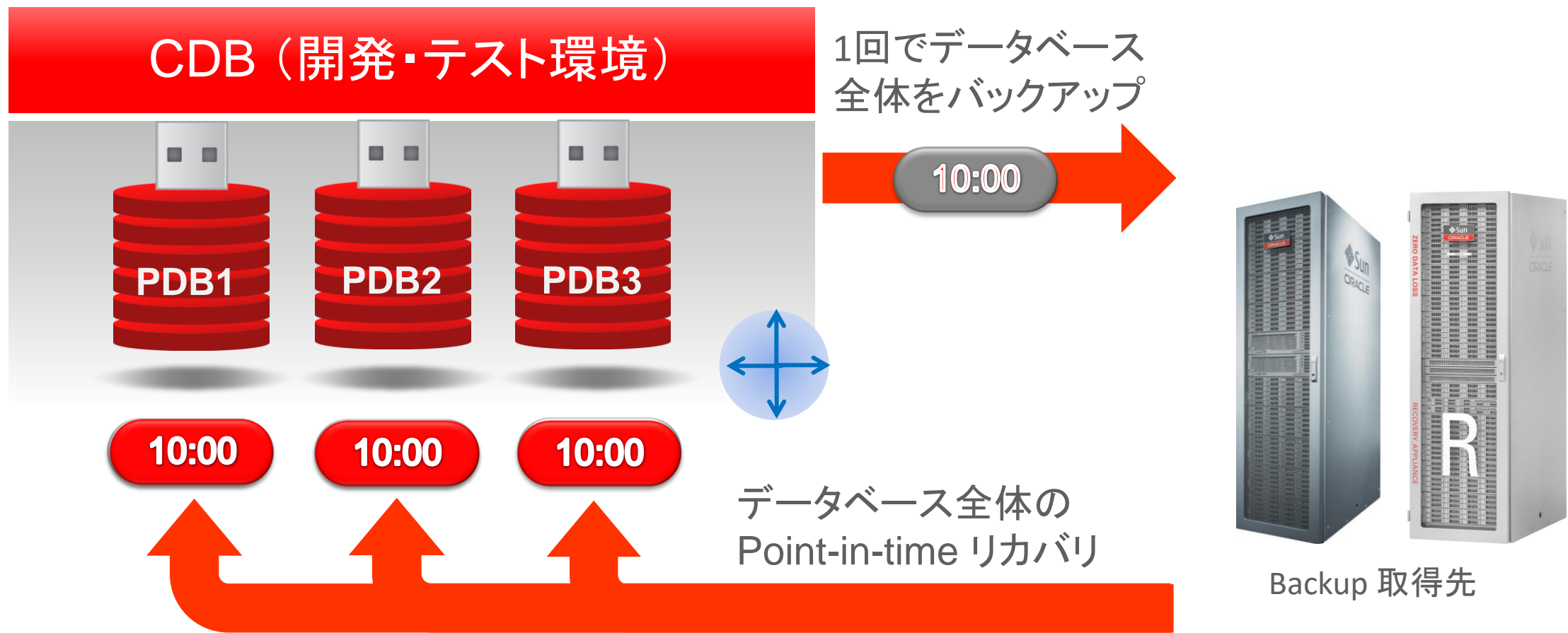
～今宵のテーマ:「マルチテナント・アーキテクチャ」を語ろう～

- 本当に必要としている技術やTipsについて、熱く語り合いましょう！
 - 今宵のテーマは、技術者の皆様から要望が高かった「マルチテナント・アーキテクチャ」を語ろう
 - マルチテナント・アーキテクチャを採用した時に気になる点や実際に得られる効果
- ファシリテーター: 田子 得哉
 - 日本オラクル株式会社
クラウド・テクノロジー事業統括
Database & Exadataプロダクトマネジメント本部
本部長

Topic#1

Multitenant 環境ではバックアップ、
リカバリがどう変わるのか?

PDB環境でのバックアップ・リカバリ



PDB作成後には必ずバックアップを取得して下さい

- 「シードからPDBを作成する方法」「ローカルPDBのクローニング」の手順の中に下記の記載があります

– https://docs.oracle.com/cd/E57425_01/121/ADMIN/cdb_plug.htm#CEGHFAGA

4. PDBをバックアップします。

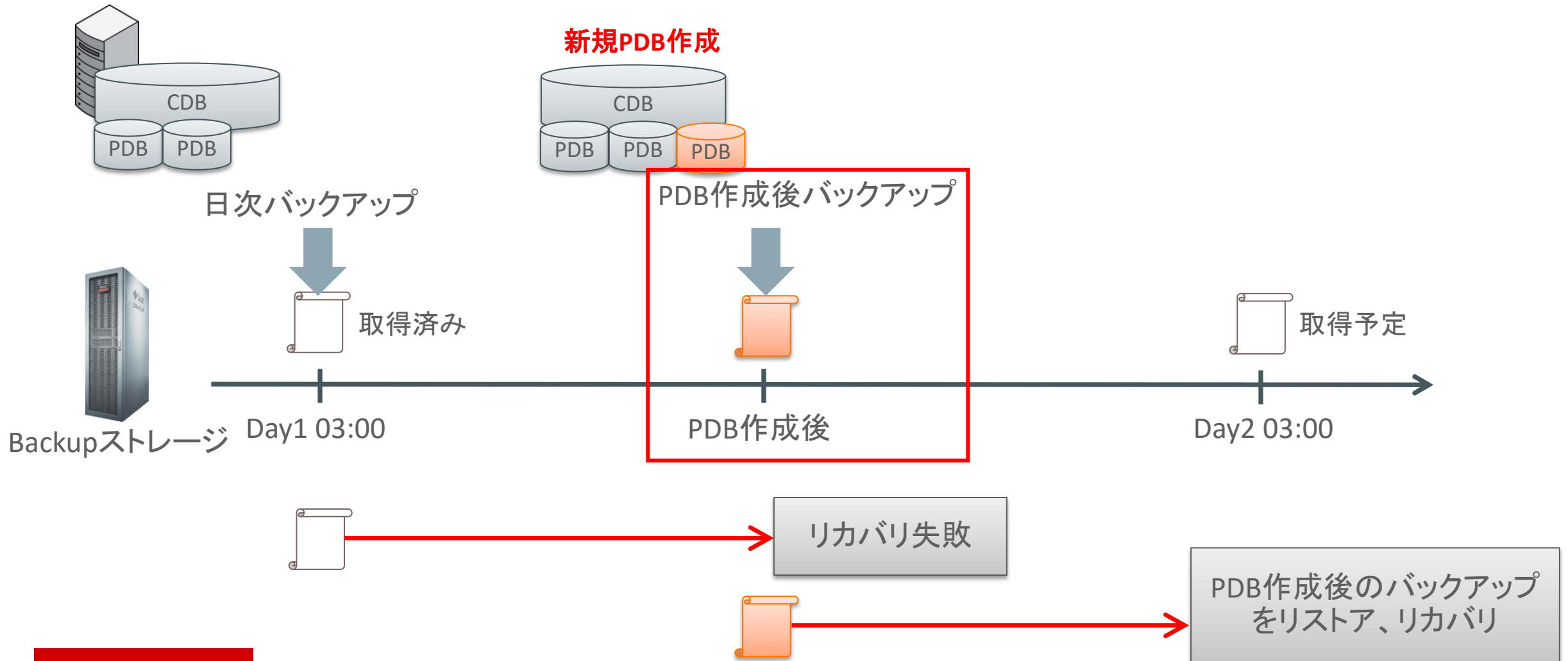
PDBは、バックアップしないとリカバリできません。

PDBのバックアップの詳細は、『*Oracle Databaseバックアップおよびリカバリ・ユーザーズ・ガイド*』を参照してください。

- PDBの作成/複製を跨ったリカバリは出来ません

```
RMAN-20505: create datafile during recovery
RMAN-11003: failure during parse/execution of SQL statement: alter database recover
logfile
'/u02/app/oracle/fast_recovery_area/dbm/DBM/archivelog/2016_09_28/o1_mf_1_1_cypl3c8v_.arc'
ORA-00283: recovery session canceled due to errors
ORA-01244: unnamed datafile(s) added to control file by media recovery
ORA-01110: data file 25:
'/u02/app/oracle/oradata/DBM/3D8ADA5071F77FE2E0533897B90AF336/datafile/o1_mf_system_cypkzghz_.dbf'
```

PDB作成後には必ずバックアップを取得して下さい



Topic#2

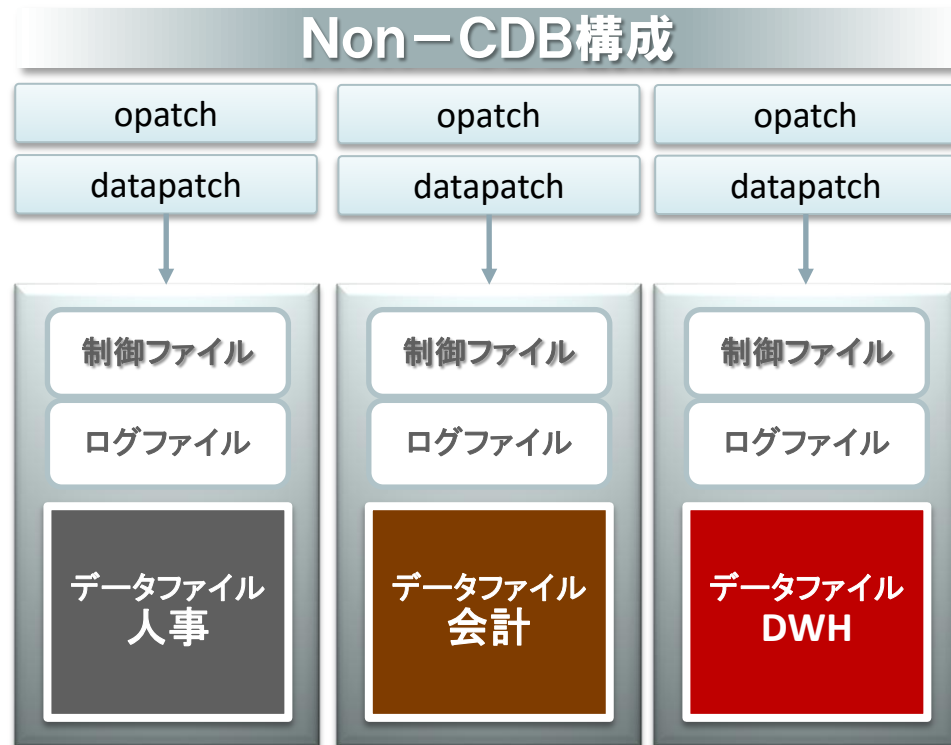
Multitenant環境ではパッチ適用は
どうなるか?

Multitenant環境ではパッチ適用はようになるか？

パッチ適用は以下の2つのコマンドを実行する必要がありますが…

```
$ opatch apply
```

```
$ datapatch -verbose
```



環境毎に opatch、datapatchの実行が必要



1回の実行で全PDB環境に適用可能

①パッチ適用におけるありがちなミス

- opatchは実行したけど、datapatchの実行をし忘れた・・・
- パッチ適用の横展開を行ったが、一部環境だけ実行できていなかった・・・
- パッチレベルの違う環境が多数存在し、どの環境で何のパッチが当たっているか確認しないと分からない・・・

12c環境では DBA_REGISTRY_SQLPATCHを検索する事で datapatchが実行されているか確認可能です。

例:PSU 12.1.0.2.161018を適用し、ロールバックした場合

```
SQL> select patch_id, flags , action, status, description, action_time from dba_registry_sqlpatch;
```

PATCH_ID	FLAGS	ACTION	STATUS	DESCRIPTION	ACTION_TIME
24006101	NB	APPLY	SUCCESS	Database Patch Set Update : 12.1.0.2.161018 (24006101)	20161213 19:28:16
24006101	NB	ROLLBACK	SUCCESS	Database Patch Set Update : 12.1.0.2.161018 (24006101)	20161213 21:40:59

②パッチ適用における Multitenant環境のメリット

- opatch、datapatchが1回の実行で全環境に適用される為、作業負荷および datapatch 実行し忘れなどの作業ミスが削減可能
- 万が一 datapatch実行時に一部のPDBをオープンしていなかったとしても、未適用のPDBが次回起動時に制限モードでオープンされる為、datapatch未実行である事が気づける

パッチ適用関連の公開資料

- Datapatch: データベース 12c パッチ適用後の SQL 自動化 (Doc ID 1950946.1)
- 12.1.0.2 の異なる PSU が適用された環境での Unplug/Plug (Doc ID 2108353.1)
- Datapatch 実行後に PDB プラグインまたはクローン DB が PDB_PLUG_IN_VIOLATION で違反を返す (Doc ID 1931071.1)
- 12.1: DBCA (データベース作成) は、「datapatch」を実行しません (Doc ID 2150789.1)

Topic#3

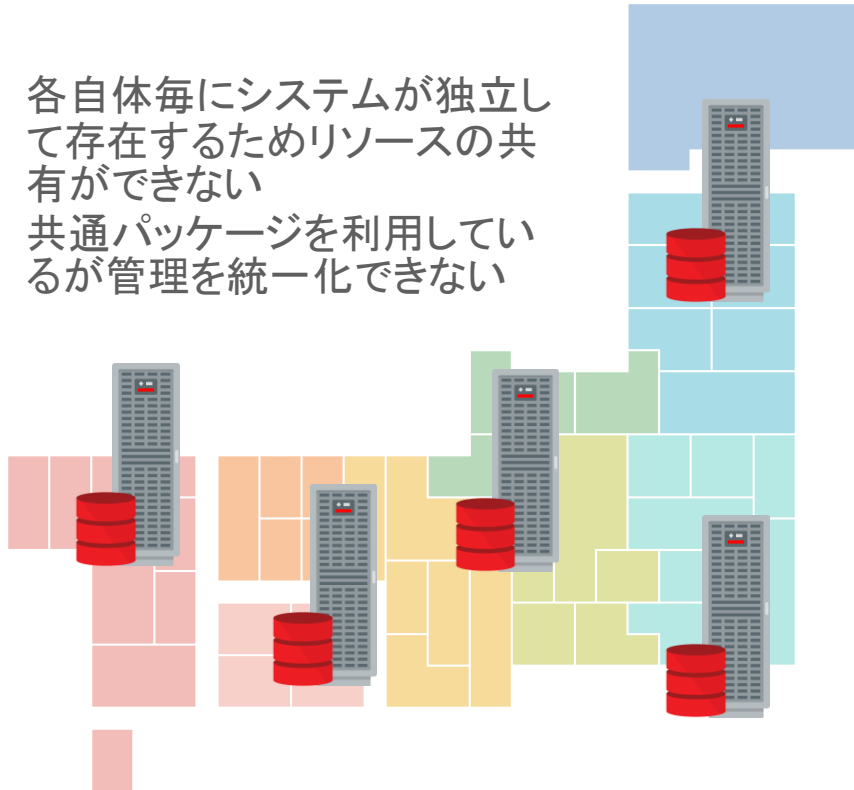
*Multitenant環境の運用で
気を付けるべき点や、便利機能、
実施すべき設定などを知りたい。*

某公共案件でのMultitenant化

- Multitenantによるリソース利用効率化と基盤コスト／管理コストの削減

Before

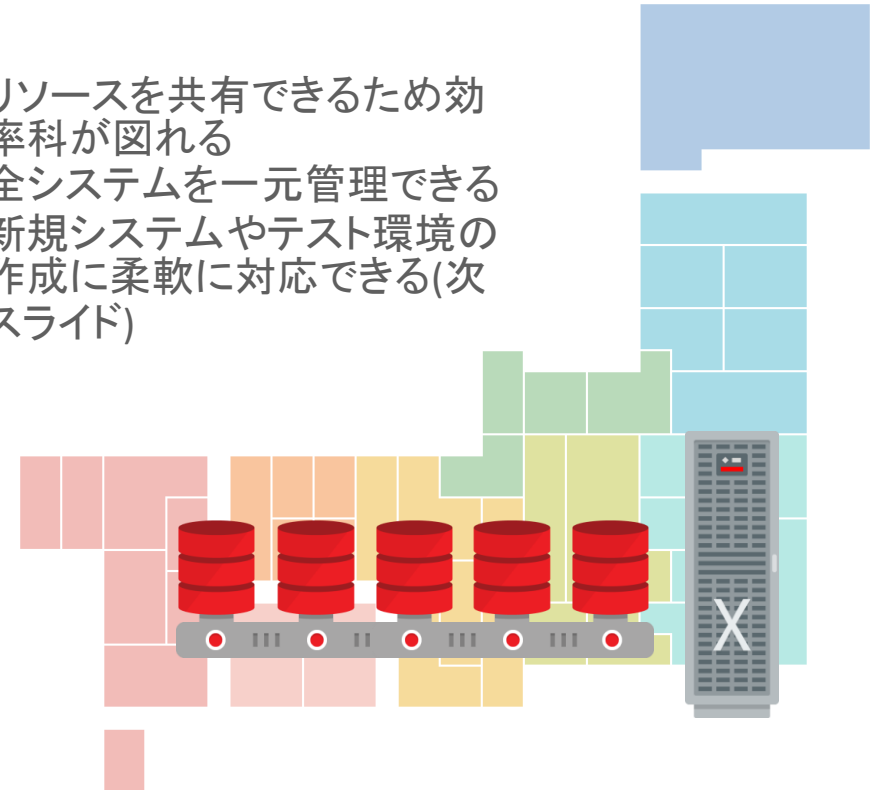
- 各自体毎にシステムが独立して存在するためリソースの共有ができない
- 共通パッケージを利用しているが管理を統一化できない



Multitenant化

After

- リソースを共有できるため効率化が図れる
- 全システムを一元管理できる
- 新規システムやテスト環境の作成に柔軟に対応できる(次スライド)



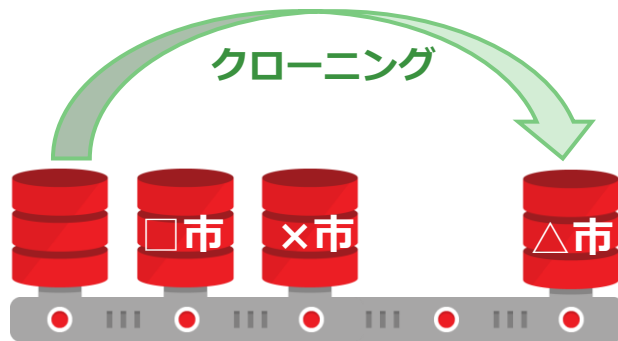
Multitenant便利機能

- PDBクローン

- 新規システム追加



〇県△市のシステムを
新規に作りたい



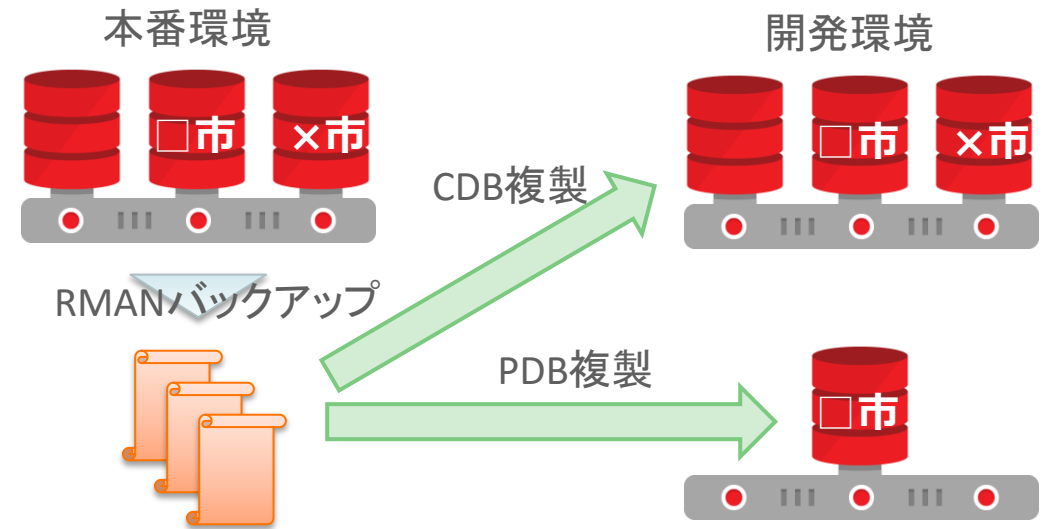
システムのベースとなるPDBを複製(クローニング)する

- 共通パッケージ、共通スキーマ、共通データを含む
- システムのテンプレートとなるPDB

- バックアップからのCDB/PDB複製

- 開発環境作成

- レポーティング環境作成



柔軟な複製が可能

- CDB単位 or PDB単位
- PITRによって特定断面のPDB複製も可能

Multitenant考慮点

• リソース・マネージャの利用

– CPUリソース制御

- CDB間 → 全CDBにCPU_COUNTを設定 or リソース利用を制限したいCDBのみCPU_COUNTを設定
- PDB間 → サービス毎にUTILIZATION_LIMITで制御(12.2からPDB単位でCPU_COUNTを設定可能)

– メモリー制御

- CDB間 → SGA_TARGETなどを適切に設定する
- PDB間 → 12.2からPDB毎に制御可能

• 初期化パラメータ考慮点

– db_files を大きくしておく(PDB毎にデータファイルが作成されるため)

– processesも考慮(各PDBのセッション数を合計)

- ### – _datafile_write_errors_crash_instance=false設定 (PDBのデータファイルへの障害発生時にインスタンスダウンを回避させるため Multitenant best Practice and Known issues (ドキュメントID 1604135.1))

Topic#4

*Multitenant*を社内で上申する際、
経営層やマネージメント層にアピ
ールできるポイントは？
(コスト・セキュリティ・可用性等)

事前に頂戴したご質問と本日の流れ

上申資料の参考として下さい



マルチテナント構成とする理由・目的を教えてください。

⇒ ① 主なユースケースと目的を説明します

Multitenantによるデータベース統合の効果がどの程度か興味がある。

⇒ ③ 事例を用いて、課題、効果、DB統合推進ポイントを紹介します

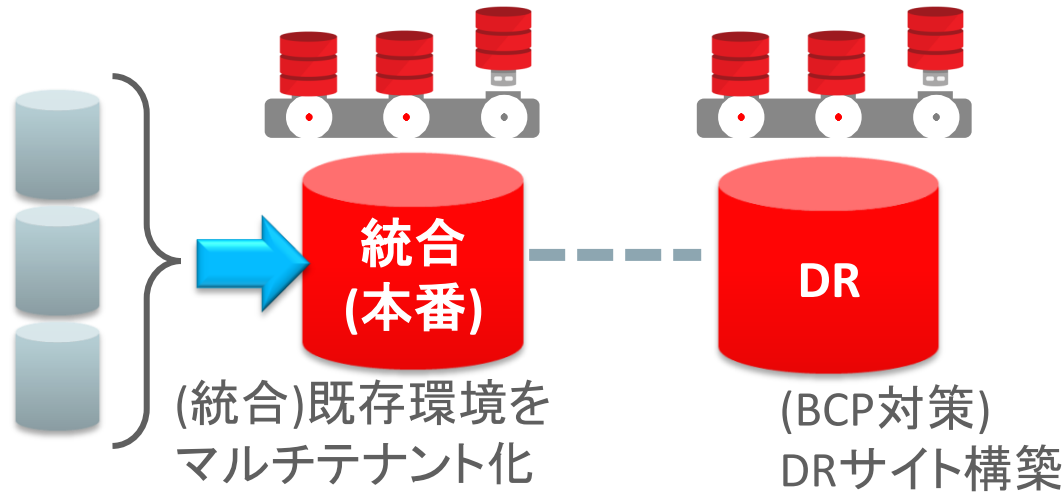
Multitenantによる統合効果が謳われていますが、多くの重要なDBにおいては、統合は行われず、単独で1PDB構成になると思います。

DB統合において気を付けるポイント等を知りたいです。

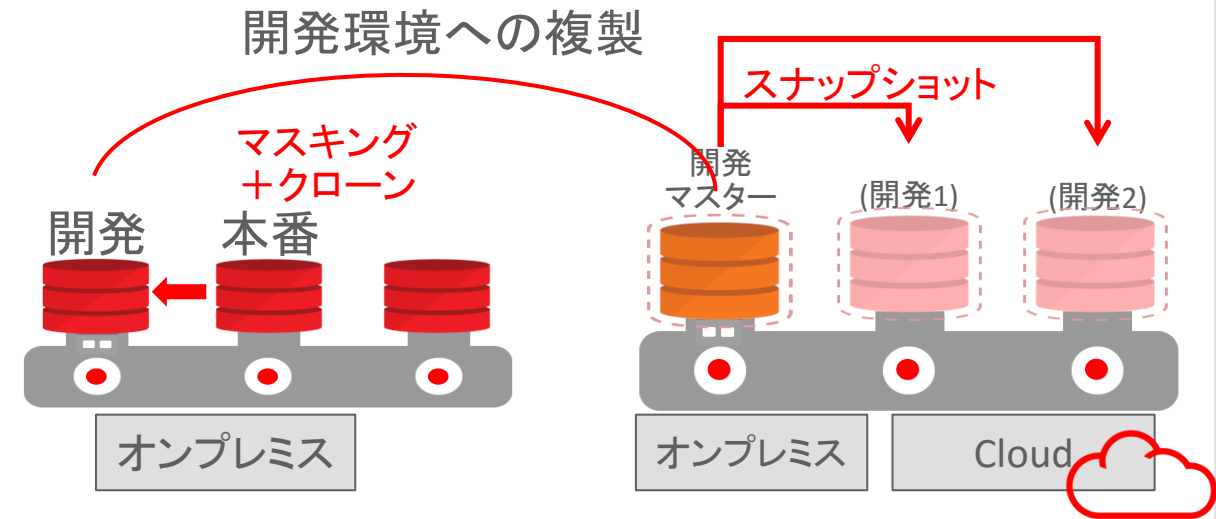
⇒ ② DB統合方式のメリ・デメを説明します

マルチテナントの代表的なユースケース

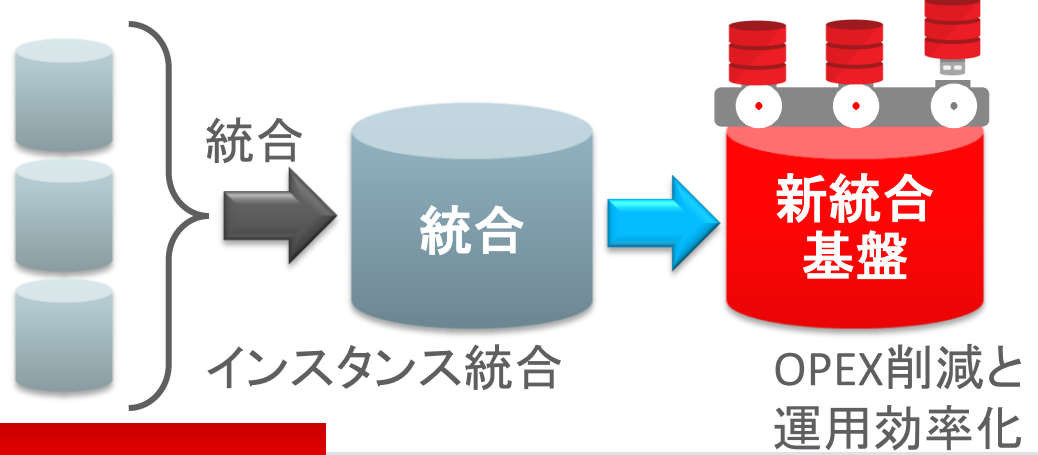
① 既存システムの統合基盤(CAPEXとOPEXの削減)



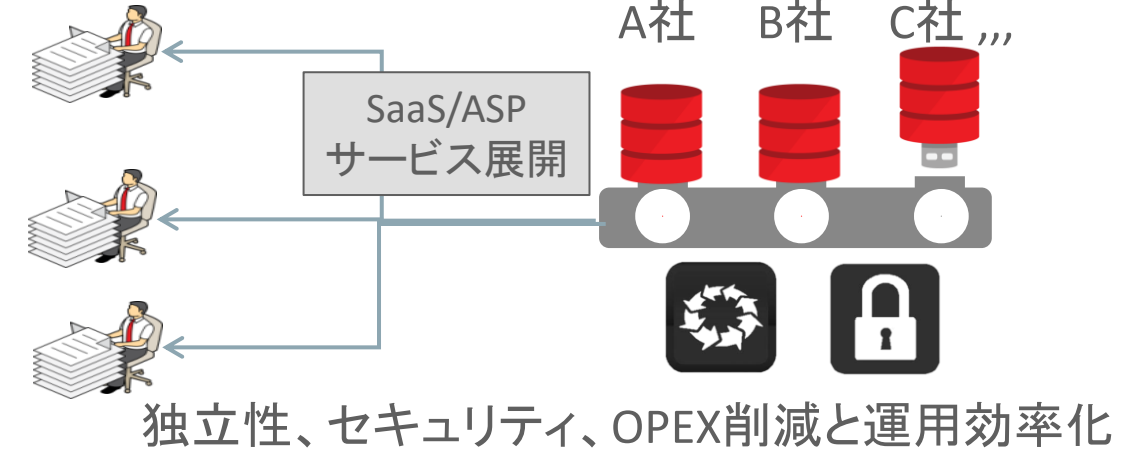
③ 開発・検証環境のアジリティ向上



② 統合システムの更改(12cにアップグレード)



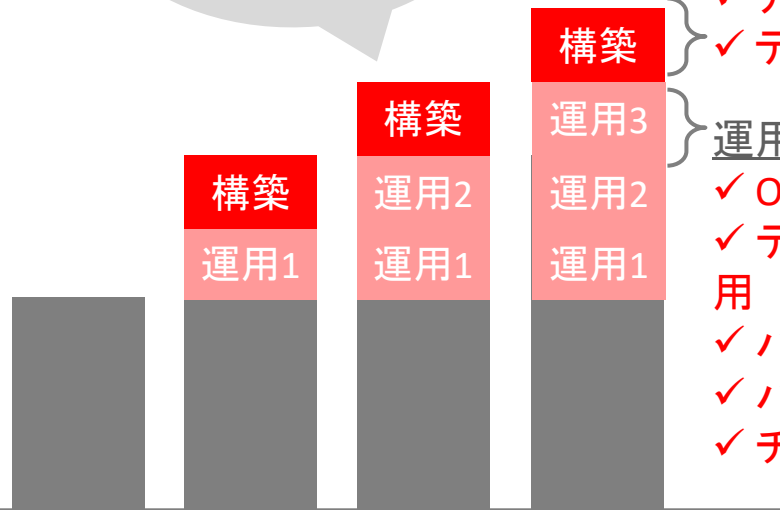
④ SaaS/ASPサービスの基盤としての活用



DB統合における従来のアプローチ

VMによるサーバ統合では、DB運用コストは下がらない

環境数と運用コストが比例して増える



仮想化環境での運用イメージ

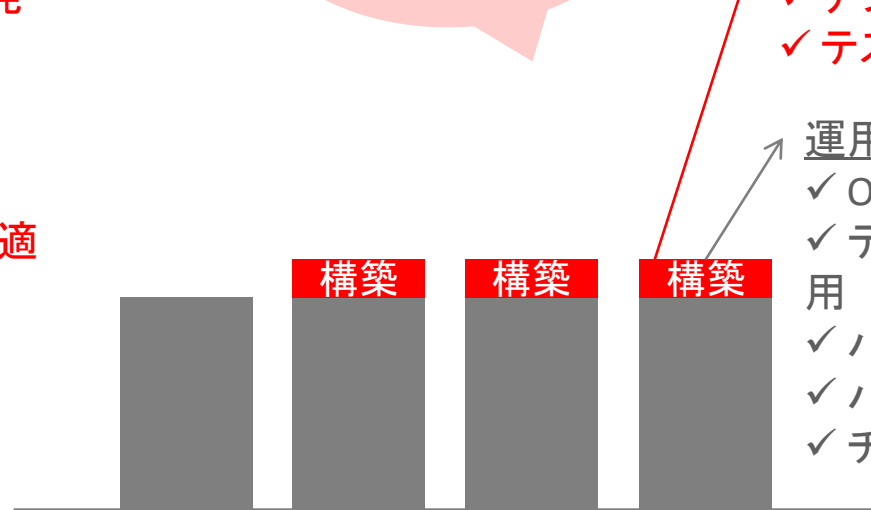
構築費の内訳

- ✓ 機器調達 (本番・テスト)
- ✓ 環境構築
- ✓ アプリケーション開発
- ✓ テスト

運用費の内訳

- ✓ OSパッチ適用
- ✓ データベースパッチ適用
- ✓ バックアップ
- ✓ バッチ処理監視
- ✓ チューニング

共通化部分の運用・開発費を圧縮



理想的な統合プラットフォームの運用イメージ

構築費の内訳

- ✓ 機器調達 (本番・テスト)
- ✓ 環境構築
- ✓ アプリケーション開発
- ✓ テスト

運用費の内訳

- ✓ OSパッチ適用
- ✓ データベースパッチ適用
- ✓ バックアップ
- ✓ バッチ処理監視
- ✓ チューニング

DBサーバの共有方式 — 各方式の特徴

集約密度 分離性

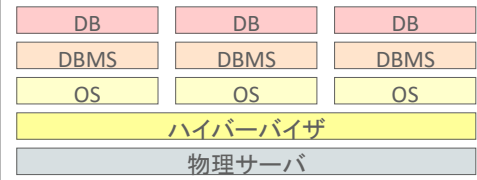
低

高

高

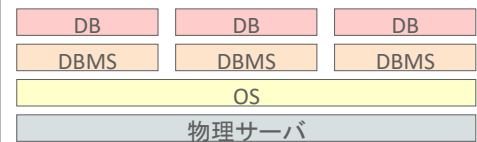
低

①ハイパーバイザによるサーバ仮想化 (IaaS型資源共有)



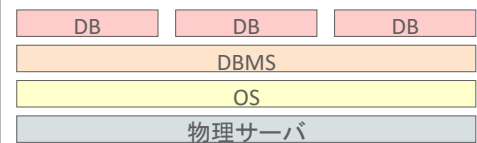
•ハイパーバイザによって物理サーバ上に複数の仮想マシンを作り、その上で各業務DBごとにOSとDBMSを並行実行する方式。

②DBインスタンス分割



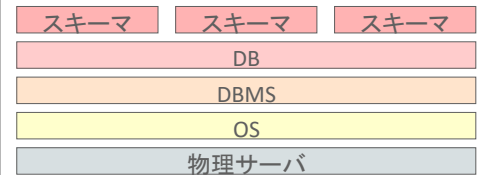
•1つのOS上で業務DBごとのDBインスタンスを並行起動する方式。

③DBマルチテナント (PaaS型資源共有)



•1つのDBMS上で、複数の業務DB(テナント)を実行する方式。各テナントには、隔離されたDBサーバ環境が仮想的に割り当てられる。

④DBスキーマ分割



•1つのDBインスタンス上で業務DBごとのスキーマを並行実行する方式。

DBサーバの共有方式の比較

①ハイパーバイザによるサーバ仮想化(IaaS型資源共有)

- 多種多様な小規模システムの集約には適しているが、運用保守効率化によるコスト削減に限界がある点、I/Oオーバーヘッド顕在化の可能性がある点等から、**比較的規模の大きいOracle Databaseの集約には推奨しない。**

コスト	非機能要件	運用上の自由度
<ul style="list-style-type: none"> • 物理サーバ台数は削減できるが、管理対象のOSやミドルウェアの個数は削減できないため、運用保守効率化によるコスト削減には限界がある。 • 製品によってはハイパーバイザのライセンス/保守費用が発生する。 	<ul style="list-style-type: none"> • 信頼性や性能面で仮想マシン間の分離性が高い。 • 仮想マシン毎にCPUとメモリの割り当て制御が可能。 • 実績が豊富であり、信頼性の面では特に問題がない。 • 大量のI/Oを伴うバッチ処理等では、I/Oオーバーヘッドが顕在化する可能性がある。 	<ul style="list-style-type: none"> • 多種多様なOSやDBMSからなる既存のソフトウェア資産を、変更なしにそのまま集約することができる。 • 業務DBごとにOS単位での再起動が可能のため、運用上の自由度は最も高い。

DBサーバの共有方式の比較

②DBインスタンス分割

- 業務DBごとにDBのバージョンアップやパッチ適用を独立して実施したい等、運用の独立性が求められる場合に適している。例えば、**複数の異なる標準システムを共通のDBサーバに集約**する場合等。

コスト	非機能要件	運用上の自由度
<ul style="list-style-type: none"> • 管理対象のOS数を削減することで、運用保守の効率化によるコスト削減効果が期待できる。 • サーバ仮想化に比べて、資源オーバーヘッドが小さいため、より集約度を上げることが可能。これによる、サーバとソフトウェアのコスト削減効果が若干期待できる。 • Oracle Databaseの追加のライセンス／保守費用が不要。 	<ul style="list-style-type: none"> • 信頼性や性能面でインスタンス間の分離性が高い。 • DBインスタンス毎にCPUとメモリの割り当て制御が可能。 • 実績が豊富であり、信頼性の面では特に問題がない。 • 性能オーバーヘッドは、サーバ仮想化とマルチテナントの中間。 	<ul style="list-style-type: none"> • 業務DBごとにDBMS (DBインスタンス) 単位での再起動が可能。

DBサーバの共有方式の比較

③DBマルチテナント(PaaS型資源共有)

- 業務DBの独立性を保ちつつ、**集約密度を高めるとともに運用保守を一元化したい場合に最適な方式。**

コスト	非機能要件	運用上の自由度
<ul style="list-style-type: none"> • 物理サーバ数だけではなく、OSとDBMSの数を削減することができるため、運用保守効率化によるコスト削減効果が高い。(例えば、全テナントDBのバックアップを一括して行える、パッチを一括して適用できる等) • DBバージョンアップやサーバ更改等の際のDB移行が容易になるため、移行期間短縮・コスト削減が可能。 • 資源オーバーヘッドが小さく、高密度集約が可能。これにより、サーバとソフトウェアのコスト削減効果が期待できる。 • Oracle Database EEとMultitenantオプションのライセンス／保守費用が必要になる。 	<ul style="list-style-type: none"> • 信頼性や性能面でテナント間の分離性が高い。 • テナント毎にCPUの割り当て制御が可能。 • 実績が豊富であり、信頼性の面では特に問題がない。 • 性能オーバーヘッドが小さい。 	<ul style="list-style-type: none"> • 全テナントのDBのバージョンとパッチレベルを統一する必要がある。(②と組み合わせて、1つの物理サーバ上でバージョンやパッチレベルの異なる複数のDBインスタンスを起動し、それぞれをマルチテナント構成にすることは可能。さらにテナントDBをDBインスタンス間で移動させることも可能。)

DBサーバの共有方式の比較

④DBスキーマ分割

- スキーマ間の分離性が低いため、複数の既存の個別システムのDB集約や運用の独立性を求められるDBの集約には不向き。

コスト	非機能要件	運用上の自由度
<ul style="list-style-type: none"> OS、DBMS、DBインスタンス等の管理対象の数を最小限にすることが可能なため、運用保守の効率化によるコスト削減効果が期待できる。 資源オーバーヘッドが最も小さいため、より集約度を上げることが可能。これによる、サーバとソフトウェアのコスト削減効果が最も期待できる。 Oracle Databaseの追加のライセンス／保守費用が不要。 	<ul style="list-style-type: none"> 信頼性や性能面でスキーマ間の分離性が低い。 性能オーバーヘッドが最も小さい。 	<ul style="list-style-type: none"> 業務DB間の分離性が低い。具体的には、オブジェクト名等の重複が許されない、独立して起動・停止ができない、パラメタ等の個別設定ができない等の制限がある。

リコー様事例

UBS様でのIaaSアプローチによる統合と課題

- UBS様の当時の状況
 - 1,500 の物理サーバー
 - 2,600 の仮想サーバー
 - 計4,100のサーバー群
 - 年5%の割合でサーバー数は増加
 - 3ペタバイトのデータ量
 - 年30%の割合でストレージ容量は増加
 - 稼働するOracle Databaseは9000個



俊敏性の欠如

システム構築のリードタイムが長く、業務側が求める**スピード感**に**応えられない**

コストの増加

アーキテクチャの複雑化と運用保守作業の属人化により、**維持コストが増大**

リスクの増大

システム老朽化と可用性欠如により、**システムリスクが増大**

UBS様が目指した最適な共通基盤像

- 業務側の要望に応える弾力性と拡張性
 - 素早いDBプロビジョニング
- コストの最適化
 - コンソリデーションによる最適な資産活用
 - 標準化、自動化、セルフサービス化による運用作業効率化とコスト削減
 - 使用量に応じた利用部門への課金
- リスク低減
 - SLAベースでの事業継続性、可用性、セキュリティ・レベルの向上

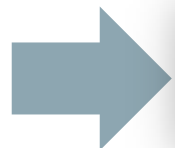


プライベートDBaaS基盤

プライベートDBaaSを実現するために必要だった事

UBS様がOracle Databaseに求めた条件

- アプリケーションに対して透過的であること
 - 統合したことでアプリケーションへの影響を最小限にする
- データベースのメタデータでの競合に対処できること
 - アプリケーション間での競合を減らし、統合時のアプリケーション変更コストを低減する
- より容易なリソース管理やワークロード管理を行えること
 - CPUやメモリだけでなく、I/Oやネットワーク帯域の層に関しても管理を行いたい
- 最小限の労力でインスタンスの移動を行えること



Oracle Database 12c で マルチテナント・アーキテクチャ として実装

UBS様が実現を目指す共通基盤

マルチテナント・アーキテクチャを核としたプライベートDBaaS基盤

[従来の共通基盤]

- 1,500 の物理サーバー
- 2,600 の仮想サーバー
- 計4,100のサーバー群
 - 年5%の割合でサーバー数は増加
- 3ペタバイトのデータ量
 - 年30%の割合でストレージ容量は増加
- 稼働するOracle Databaseは9000個



[新たな共通基盤]

- 約200 の物理サーバーに9000のOracle Databaseを統合する
 - Exadata : 57台
 - ODA : 114台
 - SPARC T4 : 51台

UBS様が共通基盤選定に際して検討した選択肢

業務側の要望に応えられない現状に対して、より持続可能で、自動化されたサービスベースのアプローチを検討した

検討した選択肢

アプローチ

インパクト

現状維持

- 保守切れ対応プロジェクトに毎年追われ続ける
- 限られたツールで個別パッチに対応
- アプリケーションごとに個別ハードウェアを用意
- 遅々とした標準化

- 保守切れ対応が課題として残り続ける
- パッチの脆弱性
- 巨大な非本番環境の維持
- 多様なインフラに起因するダウンタイム

DBaaSの
独自構築

- コモディティのサーバ、ストレージ、DB、Hypervisor、ネットワーク、管理ツールを別々のベンダーから調達し、自社でソリューションをエンジニアリング・構築する
- UBSの運用環境にインテグレートする

- セルフサービスと自動化を備えた低コストのマルチテナント・ソリューション
- 自社による大規模なエンジニアリング・サポートとインテグレーションが必要
- インテグレーションとトラブル・シューティングに責任を持つベンダーを確保し続けるのが困難

DBaaSの
調達

- 単一のベンダーからエンジアド・ソリューションとマイグレーション・サービスを調達する
- UBSの運用環境にインテグレートする

- 多様化やエンジニアリングとサポートのカスタム化を排除した標準的で目的にフィットしたアプライアンスによって、独自構築と同じ効果を得られる
- 多様なサービスのSLAベースでの管理

UBS様の考えるDBaaS によってもたらされるベネフィット

オンデマンド セルフサービス

- セルフサービス・インターフェイスを通じて作業を完結できる(作成、クローニング、廃止)
- 変更管理と予算承認が完了していれば、直ちにリクエストが処理される
- ネットワーク、ストレージ、OS、DBの全処理を単一リクエストで依頼可能

迅速性・弾力性

- 資源(CPU, メモリ, ストレージ)の追加・削除を要求
- 変更が承認されれば、動的に更新される
- 事前承認に対して有効な変更タイプを選択(大量処理能力)

サービス利用の 測定

- 各DBが消費する資源に応じて課金される

改善された監査と セキュリティ

- 主要DBに対するセキュリティ・レベルが向上
- システム管理者から業務データが保護される
- 業務データに対する従来より高度なアクセス制御が提供される

DB統合推進(Multi-tenantアーキテクチャ導入)ポイント

ポイント

- ビジネス上の課題を明確に定義してから着手すること
- 関連する部門の利害関係者に相談すること
- 組織に合ったロードマップと導入戦略を確立すること
- サービスカタログを作成してメリットを最大化すること



次回予告

Technology Night 第6弾

会社帰りに参加できる夕方開催セミナー

Oracle Database Technology Night

～集え！オラクルの力（チカラ）～

データベースのバックアップ・リカバリは何が正解なのか

～Oracle Database に最適化された

バックアップ・リカバリでできること～

今宵のテーマは、「バックアップ・リカバリ」です。

バックアップは毎日行われる運用なので手間をかけずに実施したいところです。一方で、バックアップは、万が一の障害時における最後の砦で、迅速に、そして確実に戻せることが重要です。この2つを両立させるために必要なポイントについて、これまでの復習やデモを交えながらご紹介いたします。

お申し込み・詳細はこちら

2017年1月24日（火）18:45～20:15（受付 18:30より）

<http://www.oracle.com/goto/jpm170124>

オラクル データベース セキュリティ 最新情報

<https://blogs.oracle.com/sec/>

11月のTech Nightで話したこと、話しきれなかったことをBLOGで公開中！

オラクル データベース セキュリティ 最新情報

データベース・セキュリティに関する考え方、ガイドライン、製品情報、事例、セミナー等の情報サイト

« [【資料公開】 Oracle Cloud... | Main | 「もくじ」のページ](#) »

はじめに ～ データベースセキュリティの基本的な考え方

By Tomohiko_Fukuda-Oracle on 11 21, 2016

データベースのセキュリティを担保するうえで、以下の5つのポイントが重要だと考えています。

1. 共有ユーザー利用などの匿名性を排除する
2. 重要なデータは集中管理し、アクセス経路を限定する
3. 最小権限の原則に則ったアクセス制御を強制する
4. 暗号化したほうがよいか迷うものは、すべて暗号化する
5. 監査するだけでなく、監視する

1. 共有ユーザー利用などの匿名性を排除する

「もくじ」のページ

By Tomohiko_Fukuda-Oracle on 11 21, 2016

「もくじ」のページです。各エントリや外部資料へのリンクをグループ化しています。

[全般]

第1回 [はじめに ～ データベースセキュリティの基本的な考え方](#) (2016/11/21)

第2回 [Oracle Databaseのセキュリティリスクチェックツール - Database Security Assessment Tool \(DBSAT\)](#) (2016/11/21)

第7回 [守るべきデータはどこにあるのか?](#) (2016/12/8)

[ID管理・認証]

第3回 [アカウント管理の重要性](#) (2016/11/24)

第5回 [データベース管理者と認証](#) (2016/12/1)

第8回 [Oracle Databaseのパスワード格納方法](#) (2016/12/12)

第9回 [簡単なパスワードが利用されていないかどうかの確認方法](#) (2016/12/15)

[アクセス制御]

第4回 [特権管理](#) (2016/11/28)

第6回 [最小権限の原則](#) (2016/12/5)

Safe Harbor Statement

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



Integrated Cloud

Applications & Platform Services

ORACLE®